UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR

Grado en Ingeniería Informática

TRABAJO FIN DE GRADO

Aprendiendo transformaciones con SVMs

Ivo Alonzo Jara Ovkaric
Tutor: Ángela Fernández Pascual
Ponente: José Ramón Dorronsoro Ibero

JUNIO 2019

Aprendiendo transformaciones con SVMs

AUTOR: Ivo Alonzo Jara Ovkaric
TUTOR: Ángela Fernández Pascual

Dpto. Ingeniería Informática
Escuela Politécnica Superior
Universidad Autónoma de Madrid
junio de 2019

Resumen

Este Trabajo Fin de Grado trata sobre una posible solución al problema de ubicar datos out-of-sample de Diffusion Map (DM). Para esta solución se plantea el uso de Support Vector Machine (SVM) para estimar la ubicación de los nuevos datos, a partir del hecho de que tanto Support Vector Machine como Diffusion Map son métodos de kernel. Esto significa que pueden ubicar los datos en un espacio diferente al que se encuentran originalmente, ya sea para poder separarlos linealmente o para la fácil compresión de los datos. Tal que, se usarían Support Vector Machines que, por medio de regresión sobre los vectores de coordenadas de los datos originales, puedan generar modelos que estimen cada una de las coordenadas del embedding para los nuevos puntos. Para esto, se expone qué son y para qué se usan los Diffusion Maps, que es el problema out-of-sample que presentan y lo que es un Support Vector Machine, cómo aplica el truco del kernel y cómo realiza la regresión a partir de un conjunto de datos. Para comprobar esta posible solución, se han diseñado experimentos que aplicarán este procedimiento en distintos conjuntos de datos; cuyos resultados se compararán visualmente con las ubicaciones estimadas por otro método reconocido para solucionar el problema de datos out-of-sample (Nyström) y con la proximidad que presentan con los datos originales de la misma clase ubicados por Diffusion Map. Siendo que, al final del trabajo, llegamos a la conclusión de que esta solución parece ser una buena alternativa para solucionar el problema out-of-sample de los Diffusion Maps; presentando resultados similares a los obtenidos por medio de Nyström, pero más dispersos, se presume que se está evitando overfitting, y ubicando los nuevos datos cerca a los datos originales de la misma clase.

Abstract

This Bachelor Thesis is about a possible solution to the problem that Diffusion Maps have when it needs to obtain the embedding for out-of-sample data. Therefore, this work proposes the use of Support Vector Machines to estimate the embedding coordinates of the out-of-sample data, starting from the fact that both Support Vector Machine and Diffusion Maps are kernel methods, which means that they can locate the data in a different dimensional space. This way, we will use Support Vector Machines to apply nonlinear regression to the original data, so we can estimate the embedding coordinates for each out of sample point. Even more, it is explained what Diffusion Map is and which are its uses, what the out of sample challenge is, what Support Vector Machine is, how it works, how it applies the kernel trick and how it does nonlinear regression. Then, this possible solution will be applied to some experiments using different data sets whose results will be compared to the estimated location given by the Nyström formula, the usual algorithm used for the out of sample problem, and to the proximity between the out-of-sample data location and the in-sample data location of the same class. At the end, we arrive to the conclusion that the use of Support Vector Machines is a good alternative to solve the Diffusion Maps out-of-sample challenge; since, it gives us a similar output with Nyström, just a little more scattered which is presumed to be because Support Vector Machines tends to avoid overfitting, and it also locate the out-of-sample data near the in-sample data of the same class.

Palabras clave

Diffusion Map, problema out-of-sample, Support Vector Machine, reducción de dimensión, regresión, Nyström.

Keywords

Diffusion Map, out-of-sample data, Support Vector Machine, Dimensionality Reduction, regression, Nyström.

Agradecimientos

Le agradezco a mis padres, Luis y Rina, y a mi hermana, Ljelka, por darme su amor, apoyo y la oportunidad de estudiar esta carrera. Les estoy eternamente agradecido.

Le agradezco a mi tutora, Ángela Fernández, por guiarme, corregirme, apoyarme e indicarme cómo proceder durante todo el desarrollo de este trabajo. Y a José Dorronsoro, por revisar mi investigación e introducirme al uso de librerías científicas de Python. Así como al resto de profesores que han sido parte de mi formación como profesional.

Le agradezco a mis tíos, Sara y Daniel, por alojarme, apoyarme y cuidar de mi aquí en España. A mi tía Sandra, por todo el apoyo y darme el ordenador que he utilizado toda la carrera. A mi tía Estrella, por todo el apoyo y ayudarme a realizar trámites en mi país. A mi abuelo, Enrique, que me apoyo y ayudo a realizar los trámites para obtener el visado de estudios. Y al resto de mi familia, por su contaste apoyo a lo largo de la carreara.

Por último, le agradezco a todos mis amigos por apoyarme y ayudarme a lo largo de los años y durante esta experiencia.

ÍNDICE DE CONTENIDOS

INDICE DE FIGURAS

INDICE DE TABLAS

1 Introducción

1.1 Motivación

En esta memoria de trabajo de fin de grado se expondrá el trabajo realizado por parte de su autor para encontrar una posible solución al problema de datos out-of-sample que presenta el método de aprendizaje de subvariedades (manifold learning) Diffusion Maps, por medio del uso de Support Vector Machines para ubicar esos datos; así como los resultados obtenidos en los experimentos desarrollados y las conclusiones obtenidas a partir de estos.

1.2 Objetivos

El objetivo de este TFG es probar si el uso de Support Vector Machine para obtener, por medio de regresión, una estimación de las funciones usadas por Diffusion Map para reducir la dimensión de los datos in-sample en cada coordenada, es un buen método para resolver el problema de out-of-sample de los Diffusion Maps. Siendo que, por medio de los modelos obtenidos por las SVMs para cada dimensión del DM, se pueda estimar la ubicación de nuevos datos sin tener que DM volver a calcular las ubicaciones con los nuevos datos desde el inicio. Sustentándose el uso de SVMs como una solución posible, en que tanto Diffusion Maps como Support Vector Machine son métodos que utilizan el truco del kernel para poder ubicar y clasificar sus datos en pocas dimensiones de forma óptima.

1.3 Organización de la memoria

La memoria consta de los siguientes capítulos:

- **Introducción:** en este capítulo se introduce la motivación del trabajo del fin de grado, el objetivo a alcanzar y el contenido de la memoria.
- **Estado del arte**: en este capítulo se explican los métodos con los que se estarán trabajando, la teoría que hay detrás de ellos, los usos de estos, y que aportan al objetivo del trabajo.
- **Diseño**: en este capítulo se detalla el diseño de los experimentos para probar nuestra hipótesis, explicando las decisiones tomadas desde el lenguaje de programación elegido para desarrollarlos, así como las librerías utilizadas.
- **Integración, pruebas y resultados**: en este capítulo se exponen los resultados de los experimentos desarrollados para comprobar nuestra hipótesis. Los mismos que serán evaluados visualmente por medio de gráficos; comparando las ubicaciones obtenidas para los datos out-of-sample con las ubicaciones estimadas por un método reconocido para solucionar el problema out-of-sample (Nyström) y la cercanía con los datos in-sample de la misma clase ubicados por Diffusion Map.

- **Conclusiones**: en este capítulo exponemos las conclusiones a las que hemos llegado al final de este trabajo, comprobando si, producto de las pruebas realizadas, hemos alcanzado nuestro objetivo. Además de proponer futuras líneas de trabajo a partir de lo descubierto.

2 Estado del arte

En esta sección explicaremos que son los Diffusion Map y su uso en el campo de machine learning y el tratamiento de big data. También, se expondrá las dificultades que tienen los Diffusion Maps en el tratamiento de datos out-of-sample y el uso del método de Nyström como alternativa de solución de este problema. Finalmente, se pasará a explicar qué son las Support Vector Machine, cómo funcionan y por qué pueden ser una solución para el problema de datos out-of-sample.

2.1 Diffusion Map

Actualmente se dispone de grandes cantidades de información disponible, la cual desea ser comprendida y analizada para poder tomarse decisiones en función de ella. Sin embargo, esta información está compuesta de muchas características (features), que, si bien aportan a la veracidad de las muestras, dificultan la organización de esta información por las interdependencias que estas presentan. Dicho esto, existe la posibilidad de elegir unas cuantas características, las más resaltantes, y usarlas para representar la información, descartando las demás. Si bien esta es una solución aceptable, esta representación obtenida no mostraría necesariamente la estructura de los datos originales. Así pues, la mejor solución sería el usar un método para reducir las dimensiones a las más destacables, pero que a la vez mantenga las estructuras locales de los datos originales.

Para logar lo anterior, un posible método es Diffusion Map. Este es un método de reducción de dimensión cuya principal característica es que es capaz de encontrar la variedad real de la dimensión menor en la que viven los datos originales. Al reducir la dimensión se consigue que los datos embebidos en el nuevo espacio vivan en un espacio euclídeo, es decir, la distancia complicada que explicaba los datos es ahora una distancia ahora una distancia euclídea usual en el nuevo espacio. Para ello, y de forma muy simplificada, DM calcula las coordenadas del embedding como los autovectores de una matriz de kernel [1]y[2]. La función kernel representa la similitud que hay entre dos puntos del conjunto, tal que se pueda comprobar la similitud entre todos los puntos, formando una geometría local de features específicas. Ya que se parte de la idea de: “En muchas aplicaciones, la única información significativa son los valores de correlación altos” [1]. De esta forma, por medio de la obtención de los kernels de todas las muestras, se puede formar una matriz de similitud. Siendo a partir de esta matriz de similitud que se pueden realizar los cálculos de los autovalores y autovectores para ubicar a los datos en dimensiones inferiores donde se encuentra la información más significativa.

De esta manera, Diffusion Map es capaz de usar un kernel no lineal para desplegar las subvariedades (manifold) de los datos originales, alcanzando una representación en pocas dimensiones. No obstante, diffusion map es un método bastante complejo por lo que no se van a explicar los detalles matemáticos de su funcionamiento; sino que nos centraremos en su capacidad de reducir dimensiones de conjuntos de datos por medio del uso de kernels.

2.2 Datos Out-of-sample y la formula de Nyström

Uno de los problemas que presenta los Diffusion Maps, es que estos no pueden estimar las coordenadas del embedding de datos ajenos a las muestras originales usadas al momento de su creación, conocidos como datos out-of-sample. Para poder tratar estos nuevos datos, sería necesario el volver a crear un Diffusion Map que ubique los datos originales y los nuevos datos, como parte del mismo conjunto de muestra. No obstante, el análisis espectral de la matriz de transición es computacionalmente muy costoso y no solo eso, sino que el número de muestras también afecta el tamaño de la matriz de similitud y con eso el cálculo de estos datos [2]; por lo que el volver a crear un DM con todos los datos no es una solución muy viable. Otra posibilidad sería el crear otro Diffusion Map que ubique los datos out-of-sample y luego graficar ambos DM. No obstante, esta aproximación no es correcta, debido a que el diffusion map ubica los datos a partir de la muestra con la que está trabajando, por lo que el DM resultante de los datos out-of-sample puede llegar a tener una forma totalmente diferente al DM original, por lo que esta no es una solución al problema.

Una aproximación más acertada para resolver el problema de out-of-sample sería el tratar de obtener algunas de las funciones que usa internamente DM para ubicar los datos, para intentar estimar esta función se usa la fórmula de Nyström. La fórmula de Nyström es un método que aproxima las autofunciones $\varphi_j(x^{(i)})$ de un kernel positivo y simétrico a partir de los autovectores $(\varphi_j)_i$ de la matriz de muestras del kernel; tal que, estos puedan permitir el crecimiento de la matriz de similitud. Esto permite realizar la ubicación de nuevas muestras dentro del DM sin tener que volver a calcular los autovalores y autovectores de la matriz de similitud. Siendo que, en síntesis, la fórmula de Nyström consiste en hacer una media de las coordenadas del embedding de los datos originiales, ponderados por la similitud del nuevo punto con estos datos.

2.3 Support Vector Machine

Support Vector Machine es un método que permite clasificar datos por medio del uso de hiperplanos de decisión. Para esto se vale del mapeo de los datos en dimensiones superiores a las que estos se encuentran originalmente. De esta forma, con el mapeado correcto, datos de dos categorías diferentes siempre puedan ser separadas por un hiperplano [4].

Para esto se ha de elegir una función no linear que eleve nuestras muestras a un plano superior. Para esto se utiliza el truco del kernel. El kernel es una función que dados dos patrones, x y x', devuelve un número real que, como ya hemos visto en los Diffusion Maps, caracteriza a la similitud entre ellos [3]. Muchas veces se utiliza Radial Basis Functions como kernels, porque tienen buenas propiedades como, por ejemplo: al ser una función exponencial, las transformadas que obtengamos nos facilitaran el encontrar una frontera entre las clases, mucho más que si se usara un kernel lineal. De esos, el más usual es el kernel gaussiano, cuya transformada es $\Phi(x)$, tal que : $\Phi(x) = \exp(-\gamma \|x - x'\|^2)$ [9] donde $\| x - x'\|$ corresponde a la norma de x con el resto de puntos del vector kernel y gamma (γ) corresponde a que tanta influencia tendrá esta nueva feature al momento de elegir la frontera de decisión.

Una vez ubicados todos los puntos en el espacio superior, es momento de elegir el hiperplano óptimo para separar las clases. Este hiperplano corresponderá a un plano de n-1 dimensiones, donde n es el número de dimensiones del espacio superior donde se encuentran los puntos, tal que este hiperplano pueda separar las muestras de distintas clases. No obstante, existen infinidad de hiperplanos que pueden realizar esta separación, por lo cual necesitamos encontrar el óptimo entre ellos. Para que este hiperplano sea óptimo, debe permitir la mayor generalización al momento de clasificar, por lo que debe de presentar el mayor margen posible con los elementos más cercanos al hiperplano. Estos elementos conformarán el conjunto de vectores de soporte (support vector), los cuales serán los patrones de entrenamiento. En la Figura 2-1, tomada de [3], se pueden identificar los puntos que forman parte del conjunto de vectores de soporte como aquellos dentro de un círculo, mientras que los márgenes son el espacio entre los puntos del support vector de la misma clase y el hiperplano.

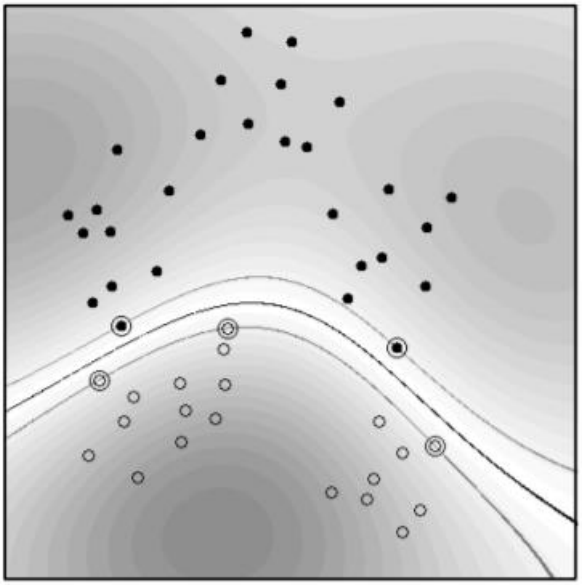

Figura 2-1: SVM con kernel exponencial

Ahora, SVM tiene una versión que permite realizar estimación por regresión. Como todas las funciones de regresión, esta trata de ubicar los datos dentro de una función al minimizar la función de coste. Así pues, por medio del uso de kernels no lineales, se crea un tubo con radio Ɛ, donde se trata de ubicar las muestras con las que se está entrenando, como se muestra en la Figura 2-2, tomada de [3], y un valor C, que será la penalización del error.

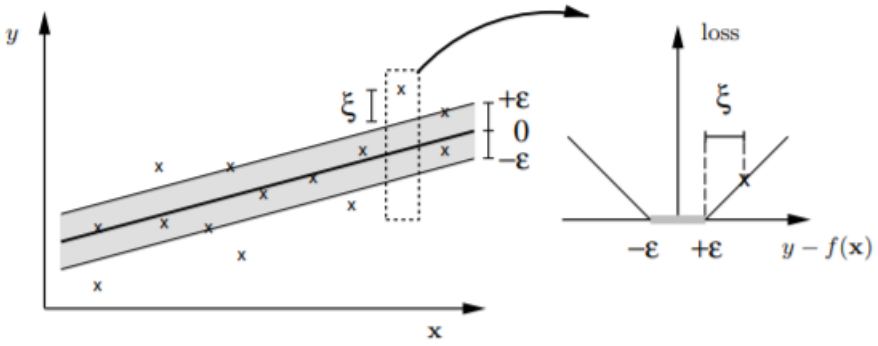

Figura 2-2: Support Vector Machine de regresión

El motivo por el cual usaremos las SVMs de regresión para tratar de solucionar el problema de los datos out-of-sample es que con ellos podríamos tratar de estimar la función Ψ que aplica internamente Diffusion Map para cada coordenada de las dimensiones reducidas. Para esto, tendríamos que entrenar una SVM de regresión para cada coordenada Ψ del DM, donde la muestras in-sample del DM serían la entrada de la SVM y el vector de coordenadas de cada dimensión del DM para las muestras in-sample sería el target. Luego de haber entrenado las SVMs, se les pedirá que cada una estime su correspondiente coordenada para los nuevos datos, las cuales luego usaremos para obtener una ubicación estimada de los datos out-of-sample.

El uso de SVM como una solución posible a este problema se encuentra en que, al igual que Diffusion Map, usa un kernel para ubicar sus datos en distintas dimensiones para poder clasificarlos. Además, al igual que Nyström, se busca estimar uno de los componentes internos que usa DM para ubicar las muestras a partir de los datos disponibles; siendo en este caso, las funciones Ψ a partir de los vectores de cada coordenada.

3 Diseño

En esta sección, se comentarán las decisiones de diseño tomadas al momento de desarrollar los experimentos. Se explicará en qué lenguaje de programación se desarrollaron y porqué se optó por el mismo; para posteriormente, detallar el diseño general de los experimentos, así como, las decisiones tomadas para obtener los mejores resultados posibles.

3.1 Elección del lenguaje de programación

Para la realización de estos experimentos se ha elegido desarrollarlos en el lenguaje Python, el cual es actualmente uno de los lenguajes de programación más utilizados en machine learning. Esto es debido a que Python nos ofrece, ya implementadas, estructuras de datos, como arrays o diccionarios, lo cual nos permite gestionar fácilmente cantidades masivas de datos. Así mismo, Python tiene la ventaja de disponer de muchas librerías especializadas que se encuentran a libre disposición de los usuarios. A continuación, se mencionarán algunas de las librerías que se han usado para la realización de los experimentos, así como las ventajas que estas proporcionan.

La primera librería que usaremos es Numpy. Esta librería nos brinda una estructura de datos llamada ndarray, la cual es un array de N-dimensiones que nos permite almacenar los datos y luego reagruparlos a nuestra conveniencia, gracias a que el tamaño de las dimensiones del array están representadas en una tupla de enteros [5]. Esto último nos permite almacenar el array de muestras, donde cada muestra es un array de características, como en una matriz, pudiendo así consultar fácilmente cada característica. Esta librería también nos brinda distintas funciones para tratar los datos almacenados, como: encontrar cuales datos cumplen con una condición, eliminar muestras y reorganizar la matriz, combinar N arrays de tamaño M para obtener uno en forma (N, M), realizar operaciones con matrices, crear arrays de datos a partir de funciones gaussianas, entre otras.

Por otro lado, empezando con las librerías especializadas en machine learning, tenemos la librería pyDiffMap. Esta librería nos brinda una implementación de diffusion maps sencilla de usar y graficar. Además, dispone de una implementación del método de Nyström que se ejecuta sobre sus mismos DM, simplificando los pasos para aplicar esta solución al problema de out-of-sample. Por último, en su documentación se incluyen ejemplos prácticos del uso de Diffusion Map, los diferentes métodos que ofrece la librería y una sucinta explicación sobre los fundamentos matemáticos detrás de los diffusion maps [6].

A continuación, tenemos la librería scikit-learn. Esta es una de las librerías más populares para machine learning; ya que no solo brinda clasificadores y estimadores; sino que también nos ofrece diferentes módulos que resultan útiles para preparar los datos antes de procesarlos, elegir los mejores parámetros de los clasificadores y comparar los resultados obtenidos [7]. Para nuestros experimentos, usaremos su implementación de SVM de regresión. Además de eso, usaremos otros módulos como: el módulo de pre-procesamiento y normalización para preparar los datos para el diffusion map, el módulo de selección de modelos para elegir los mejor hiperparámetros para las SVM, y el módulo de métricas para comparar los resultados de los diferentes modelos de SVM, ante la misma entrada y salida, para elegir el más eficaz.

Por último, tenemos la librería matplotlib. Esta librería nos permite graficar los resultados de nuestras predicciones por Nyström y SVMs de las coordenadas donde se ubicarían los nuevos datos, junto con los datos que el diffusion map ya tenga ubicados previamente.

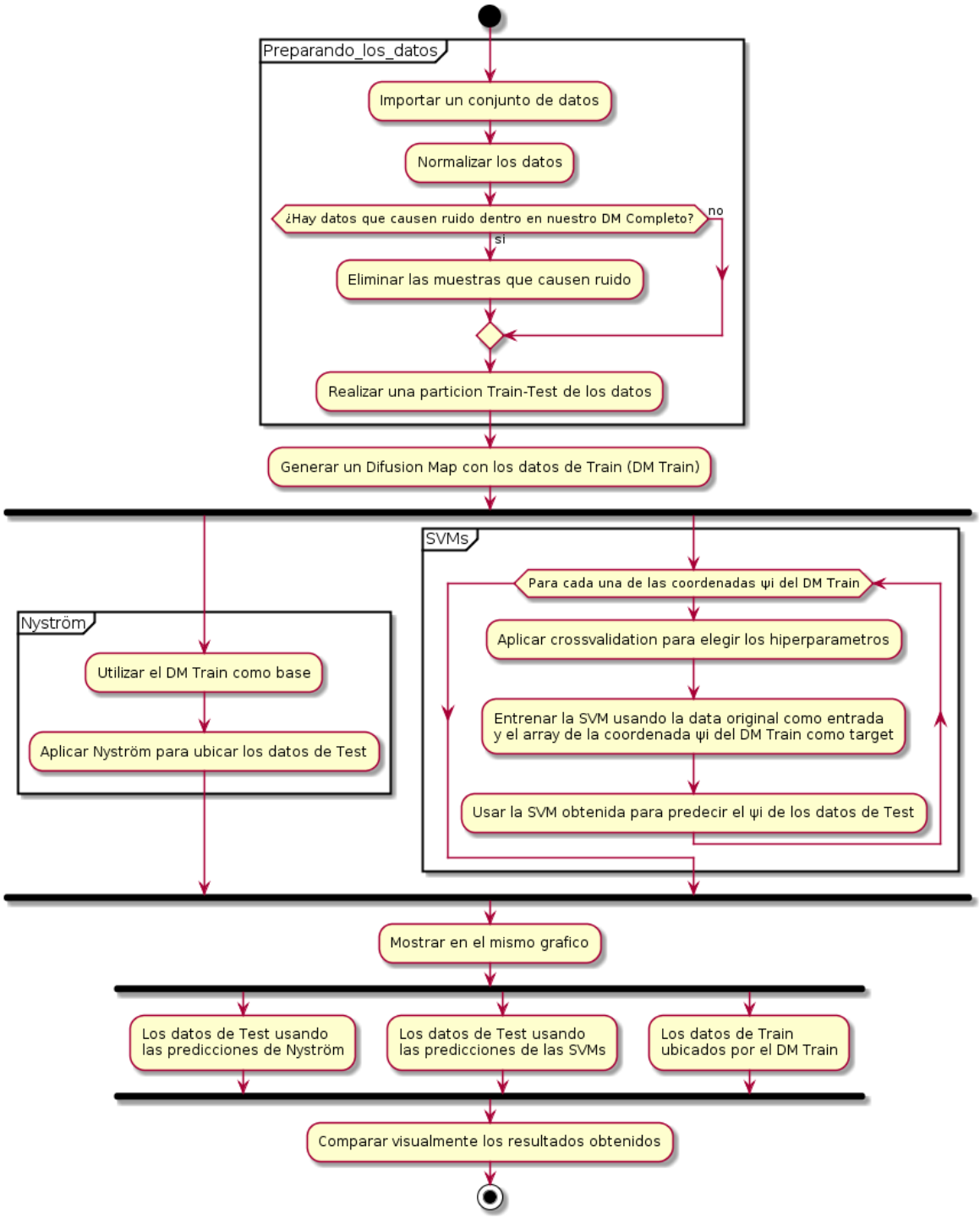

Figura 3-1: Flujograma del experimento

3.2 Diseño de los experimentos

Para probar nuestra hipótesis, se ha diseñado un experimento que comprobará si con la aplicación de SVMs se podrá solucionar el problema de out-of-sample de los diffusion maps. Para esto se trabajará con distintos conjuntos de datos, realizando una partición de las muestras en dos conjuntos diferentes, el primero será el conjunto de train y el segundo será el de test. Primero, se creará un diffusion map, el cual recibirá y ubicará todos los datos de train. Luego, para poder ubicar los datos de test aplicaremos dos aproximaciones diferentes. Por un lado, la primera aproximación será aplicar el método de Nyström, el cual es el método que se utiliza usualmente para resolver este problema, sobre el diffusion map con los datos de train. Por otro lado, la segunda aproximación será por medio de SVMs, que realizarán una regresión sobre cada una de las dimensiones ψ_i que se han obtenido en el diffusion map con los datos de train, para poder predecir cada una de las coordenadas donde se ubicarían los datos de test. Todo lo anterior mencionado se encuentra representado en el flujograma correspondiente a la Figura 3-1. A continuación, se pasará a detallar mayor profundidad las etapas del experimento.

3.2.1 Pre-procesamiento de los datos

Primero se ha de elegir el conjunto de datos con el cual trabajar. Si bien se puede trabajar con cualquier conjunto de datos, para nuestros experimentos hemos elegido, en su mayoría, conjuntos de datos de clasificación y de no muchas clases. Esto se debe a queremos que el diffusion map agrupe las muestras de la misma clase, tal que podamos diferenciar fácilmente una clase de otra y así poder comprobar visualmente que las SVMs están ubicando las muestras nuevas cerca de las muestras de la misma clase. De esta forma, se tendría una matriz con muestras y características a la que llamaremos datos y un array con las clases de cada una de las muestras, el cual será el target.

Feature	Valor más bajo	Valor más alto	Promedio
alcohol	11.03	14.83	13.000618
malic_acid	0.74	5.80	2.336348
ash	1.36	3.23	2.366517
alcalinity_of_ash	10.6	30.0	19.494944
magnesium	70.0	162.0	99.741573
total_phenols	0.98	3.88	2.295112
flavanoids	0.34	5.08	2.029270
nonflavanoid_phenols	0.13	0.66	0.361854
proanthocyanins	0.41	3.58	1.590899
color_intensity	1.28	13.00	5.058090
hue	0.48	1.71	0.957449
od280/od315_of_diluted_wines	1.27	4.00	2.611685
proline	278.0	1680.0	746.893258

Tabla 3-1: Desglose de las features del Wine Data Set

El siguiente paso será el normalizar las diferentes características de los datos. Esto se hace ya que un conjunto de datos puede presentar características que se encuentren en orden de

las centenas mientras que otras están en valores decimales, como se puede apreciar en la Tabla 3-1. Así pues, al escalar todos los vectores de características, se evita que haya alguna característica que predomine por estar en un orden mayor u otra que tenga impacto bajo por estar en un orden menor. Se suele usar el estándar de media 0 y desviación 1 (zero mean & unit variance) para normalizar los datos, el mismo que usaremos para nuestros experimentos; sin embargo, también se pueden aplicar otras formas de normalizar como unit norm.

Una vez hecho esto, revisaremos si es que entre las muestras de nuestros datos hay datos anómalos, y en el caso de que los hubiese, pasaremos a eliminarlos de las características y el target. Esto se hace para evitar que la aparición de datos aislados en los diffusion maps afecten la apreciación del resto de las muestras en el gráfico. Además, debido a la naturaleza de las SVM, estas suelen ser descartadas al momento de realizar las regresiones de los ψ, por lo que nuestros resultados corresponderán al resto de las muestras.

Finalmente, se pasará a dividir los datos normalizados y el target en un conjunto de entrenamiento (train) y otro de prueba (test). Esta división se hará tal que en el conjunto de train tengamos un 75% de los datos totales aproximadamente; mientras que el 25% restante se usarán para el conjunto de test. Esto se hace para poder probar, en condiciones reales, la predicción de estas para datos nuevos (los datos del test) a partir de los datos con los que ha entrenado (los datos del train). Es importante que la división del train y el test este estratificada, es decir, que el conjunto train tenga el 75% de las muestras de cada clase, mientras que el test tenga el resto; ya que, de no hacer así sino eligiendo un porcentaje de muestras al azar, puede ocurrir que en el test se encuentre el 99% de las muestras de una clase y el train solo tuviese un 1%, dificultando así la ubicación de esta clase. Por eso, en el caso de que los conjuntos de datos que se estén tratando vengan ya divididos en conjuntos train y test, se usarán esos mismos para el experimento.

3.2.2 Diffusion Map y Nyström

Ahora generaremos un diffusion map, al cual entrenaremos con las muestras del conjunto de train. El número de dimensiones a las que se reduzca el diffusion map dependerá del conjunto de datos con el que se esté trabajando. Así pues, usualmente se reducirá a 3 dimensiones para poder graficarse; sin embargo, puede darse el caso que se logre apreciar la agrupación de las muestras de las mismas clases con solo dos dimensiones o que este número de dimensiones no sea suficiente para apreciar la separación de clases.

Debido a las característica de los Diffusion Maps, estos no puede ubicar nuevos datos a partir de los que ya tiene embebidos, sino que tendrían que volverse a crear, ahora con todos los datos, para que estos puedan ser ubicados. Así pues, para poder predecir donde se ubicarían los datos de test, aplicaremos la primera aproximación a resolver este problema, usar el método de Nyström en el diffusion map que contiene los datos de train. Si usamos la librería pyDiffMap para crear el diffusion map de train, podremos utilizar la función Nyström que nos proporciona y obtener los vectores de cada coordenada de las muestras de test. Con estos vectores de coordenadas podremos pintar sobre el gráfico con las muestras del train, una aproximación a donde se ubicarían las muestras del test.

3.2.3 Prediciendo con SVM

Una vez obtenida la predicción por Nyström, pasaremos a aplicar nuestra aproximación de que podremos predecir la ubicación de las nuevas muestras del test (out-of-sample) por medio de regresión de las SVM sobre cada una de las coordenadas ψ del diffusion map con los datos de train.

Para esto, crearemos una SVM de regresión por cada una de las coordenadas del diffusion map. El objetivo de cada SVM es encontrar, por medio de la regresión de los datos con la coordenada del DM, la función que usa internamente el DM para situar cada coordenada. Además, como tanto DM como SVM son métodos de núcleo, nos asegura una mejor regresión con SVMs que si se hiciera con otro método. Actualmente la librería pyDiffMap solo tiene implementado un kernel gaussiano para el método from_sklearn (que construye el DM usando el objeto vecinos cercanos de Scikit-learn) y para su implementación del método de Nyström [6]; por lo que se recomienda que el kernel que usen las SMVs sea también una función exponencial ('rbf' para la implementación de scikit-learn), para que así ambos sean equivalentes.

Así pues, usaremos los datos del train, ya pre-procesados, como entrada de la SVM; mientras que usaremos el vector de la coordenada ψ_i, sin alterar, como el target de las muestras de train. Haciendo esto, las SVM habrán de estimar la función que utiliza internamente el diffusion map para ubicar las muestras en esa coordenada. Una vez con las SVM entrenadas para cada uno de los vectores ψ, pasaremos a predecir los datos de test en cada una de ellas. El resultado de cada una de las predicciones corresponderá a la coordenada de dicha muestra para el respectivo ψ; tal que, si se grafica usando los vectores obtenidos al predecir con las SVM de ψ_1, ψ_2 y ψ_3 como los ejes X, Y & Z respectivamente, se obtendrá la ubicación para dicho muestra dentro del diffusion map.

Si bien se puede usar los valores por defecto de los hiperparámetros las SVMs, es posible que estos generen modelos de diferente calidad para cada coordenada, debido a la escala en la que esta se encuentra. Esto podría ocasionar que se tenga un modelo muy malo, al punto que no sepa ubicar los datos out-of-sample para esa coordenada. Por esto, es preferible el ajustar cada una de las SVM para que estas nos den las mejores predicciones posibles. Para esto aplicaremos cross-validation a cada una de las SMV, pudiendo así elegir la mejor combinación de hiperparámetros para cada una de ellas. Los hiperparámetros que vamos a ajustar serán: C, gamma (si estamos usando kernel exponencial) y épsilon. Se recomienda probar con múltiplos y submúltiplos de 10 para cada hiperparámetro; no obstante, se tiene que considerar que a mayor número de combinaciones posibles más tardará en ejecutarse el cross-validation. Al final, elegiremos la combinación de hiperparámetros que obtengan un mejor resultado para la métrica de error cuadrático medio negativo (el negativo es debido a que se trata de SVM de regresión).

4 Integración, pruebas y resultados

En esta sección se detallarán los diferentes experimentos que se han realizado para comprobar nuestra hipótesis, siguiendo el diseño expuesto en la sección anterior. Por cada uno de ellos se detallarán: el motivo de su elección, propiedades del conjunto de datos, los hiperparámetros elegidos para las SVMs y los resultados obtenidos en cada uno de ellos. Cabe destacar que lo que se comparará en cada experimento será la ubicación de los datos de train, obtenido por el DM Train, con las ubicaciones estimadas para los datos del test, obtenido tanto por el método de Nyström como por las SVMs. El motivo de hacer esto es que las ubicaciones del DM para los datos de test no son conocidas. Además, en el caso de los ejemplos de clasificación, conocemos las clases de los datos out-of-sample; por lo que se espera que el embedding estimado para estos se encuentre cerca al resto de puntos de la misma clase del conjunto de train.

4.1 Parámetros generales

Existen algunos parámetros que tienen en común todos los experimentos, a continuación, se explican cuales son.

Para los diffusion maps:

- El parámetro algoritmo (algorithm), que es el algoritmo que usa por debajo scikit-learn para realizar nearest neighbors, donde hemos elegido 'ball_tree' por estar optimizado para grandes dimensiones [7].
- Para épsilon (epsilon) se ha elegido la opción 'bgh', la cual es una opción propia de la implementación de pyDiffMap que elige un épsilon adecuado para escalar la distancia entre las muestras [6].
- El parámetro alfa (alpha) se ha puesto a 1.0, esto es para que sea imparcial ante las diferentes densidades de muestras de cada conjunto de datos.

Para las SVMs:

- Se usa una SVM diferente para cada coordenada del DM y en cada una se aplica cross-validation 10-fold.
- Las opciones de hiperparámetros que se han probado han ido variando en múltiplos o submúltiplos de 10, esto con motivo de abarcar las mayores opciones posibles.
- La métrica usada para valorar y elegir el mejor de los modelos es el error cuadrático medio negativo, debido a que se trata de SVMs de regresión.

4.2 Experimento Swiss Roll

Este experimento utiliza el ejemplo de Swiss Roll, el cual es un ejemplo clásico para aplicar métodos de reducción de dimensiones, como diffusion map. El Swiss Roll es un conjunto de datos en 3D del cual es difícil encontrar una representación en 2D; esto es debido a que sus datos se encuentran enrollados, formando una especie rollo, razón por la cual recibe este nombre. Así pues, para obtener una representación útil en 2D de este

conjunto de datos, necesitamos un algoritmo que sea capaz de desenrollarlo [8]. En la Figura 4-1 se puede ver como se encuentra los datos enrollados en 3D; mientras que en la Figura 4-2 se muestra el Swiss Roll reducido a 2 dimensiones por el diffusion map. Como objetivo usaremos la suma de la primera coordenada al cuadrado y la segunda coordenada al cuadrado. Esto se debe a que el Swiss Roll se en posición vertical, por lo que la espiral se encuentra en el plano X-Y. De esta forma, las muestras que se encuentran al interior del Swiss Roll estarán de color rojo, mientras las que estar al exterior serán azules; con esto podemos apreciar como el Diffusion Map logro desenrollar correctamente el Swiss roll [6].

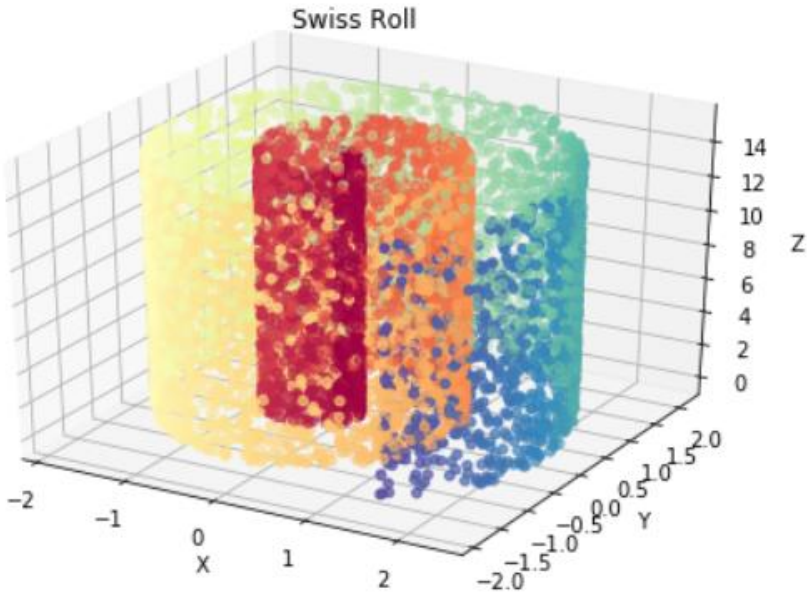

Figura 4-1: Swiss Roll en espacio 3D

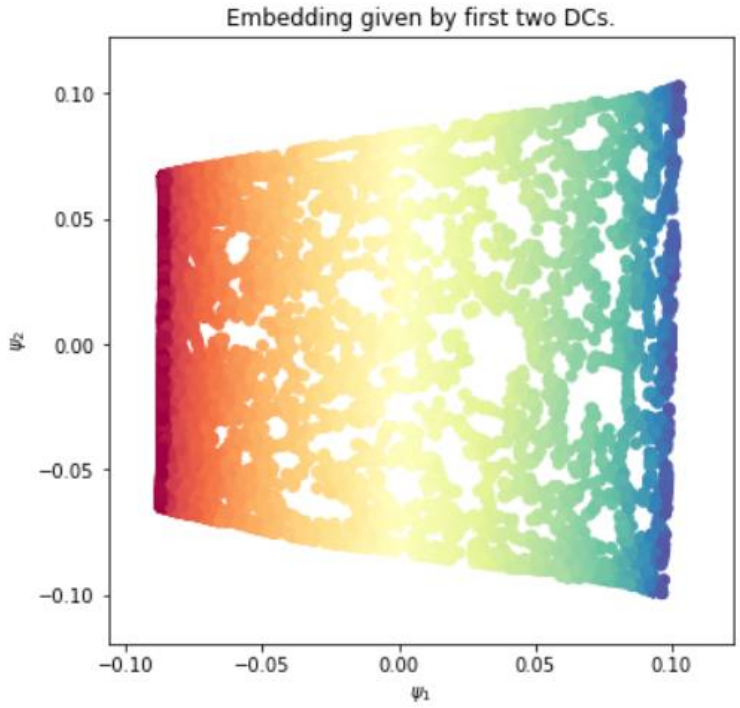

Figura 4-2: Swiss Roll desenrollado en 2D por DM

Parámetro	Valor
Nº de muestras(totales)	10000
Nº de autovectores computados (n_evecs)	10
Nº de vecinos (k)	64

Tabla 4-1: Parámetros usado para DM Train del Swiss Roll

Ya que este es un ejemplo clásico del uso de diffusion map, es un buen conjunto de datos con el cual experimentar. Así pues, no es necesario quitar datos anómalos por la naturaleza al azar de la creación del Swiss Roll, por lo que partiremos haciendo la partición train-test. Ahora, crearemos el DM Train, con los parámetros mostrados en la Tabla 4-1. En la Figura 4-3 podrá ver el gráfico resultante; cabe recordar que el target de cada muestra es la suma de los cuadrados de su la primera y segunda coordenada de los datos, formando la espiral que sale desde el centro del plano XY del Swiss Roll, y que este gráfico difiere del de Figura 4-2 por tener un K diferente y realizarse sobre un 75% de los datos originales.

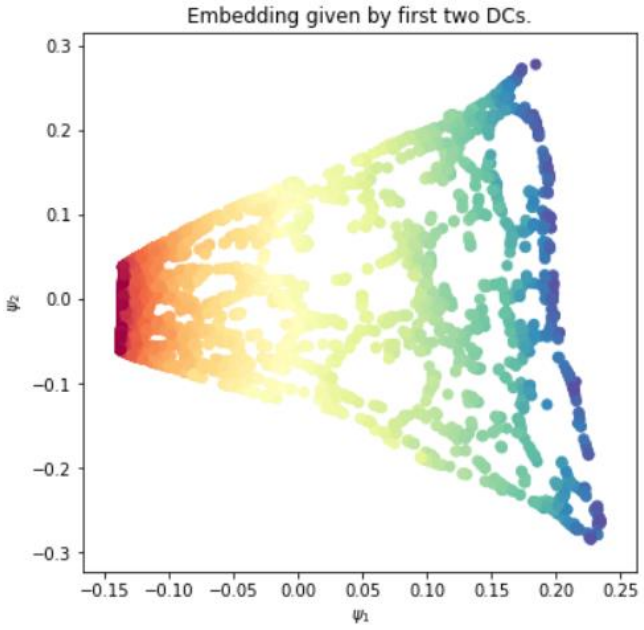

Figura 4-3: Gráfica DM Train del Swiss Roll

Coordenada DM (Ψ_i)	C	Épsilon	Gamma
Ψ_1	1000	0.01	'auto'
Ψ_2	1000	0.01	'auto'

Tabla 4-2: Hiperparámetros de las SVMs del Swiss Roll

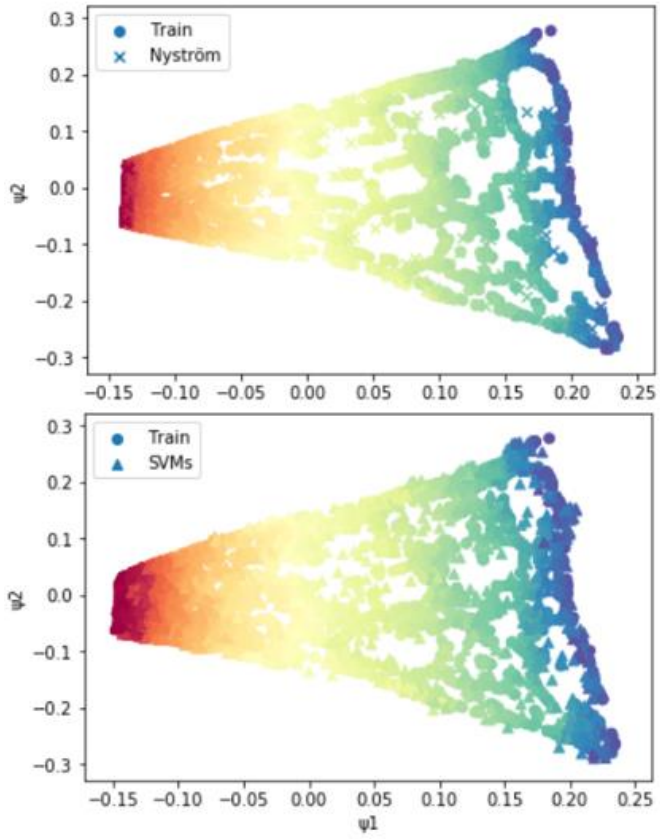

Figura 4-4: Comparación de Train-Nyström con Train-SVMs del Swiss Roll

Ahora probaremos las 2 aproximaciones para predecir la ubicación de los datos out-of-sample. Para esto aplicaremos Nyström a partir del DM Train y luego aplicaremos las SVMs para las 2 dimensiones del DM. En la Tabla 4-2 encontramos los hiperparámetros obtenidos por el cross-validation para el mejor modelo de cada una de las SVM.

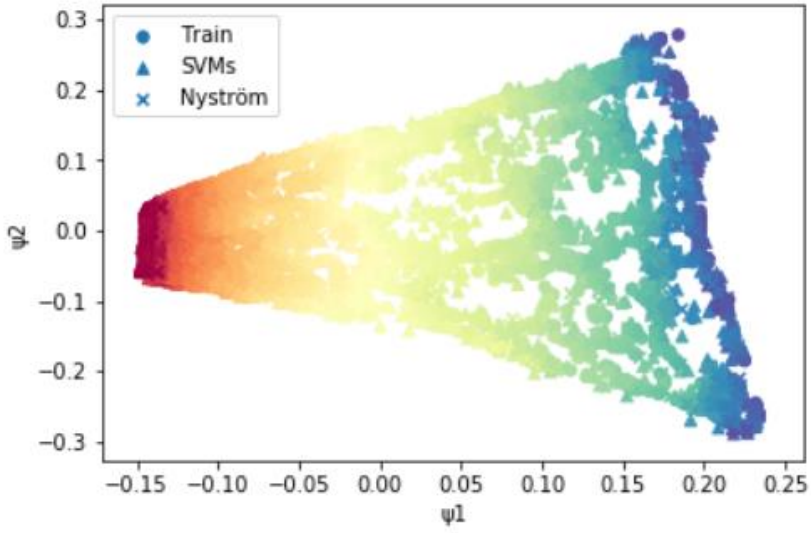

Figura 4-5: Gráfica Train-SVMs-Nyström del Swiss Roll

En la Figura 4-4 podemos ver las gráficas resultantes de pintar la gráfica del DM con los datos con: a la izquierda, el posicionamiento de los datos de test ubicados por Nyström y a la derecha, los datos de test posicionados por medio de las SVMs de regresión entrenadas con las coordenadas de Ψ_1 y $\Psi 2$ del DM Train del Swiss Roll. Si comparamos ambas gráficas, podemos notar que las predicciones de Nyström tienden más a adaptarse a la forma del DM Train, estando muy cercanas a los bordes de este. Por otro lado, en la segunda gráfica, las predicciones hechas por las SVMs si bien también siguen la forma del DM Train, estas predicciones no se pegan tanto a los bordes, sino que se encuentran más esparcidos, dando la impresión de completar posibles ubicaciones donde se situarían los datos originalmente. Esto da a entender que Nyström podría sufrir algo de overfitting al basarse en las ubicaciones de las muestras de la DM Train; mientras que las predicciones de las SVMs evitan el overfitting del train. Esto último se debe a que al tener que hacer la regresión de cada coordenada, las SVMs se fijan en las muestras que conforman el support vector para elegir el hiperplano óptimo [4]. De igual manera, ambas predicciones parecen dar un resultado correcto. Esto se puede comprobar en la Figura 4-5, donde al graficar el DM Train, las ubicaciones que nos da Nyström y las predicciones de las SVMs en la misma gráfica, se puede notar que ambas predicciones tienden a sobreponerse, salvo en los bordes de DM Train donde las SVMS se explayan algo más.

4.3 Experimento Iris

El siguiente conjunto de datos con el que trabajaremos serán el Data Set Iris. Este Data Set tiene 3 clases diferentes de plantas Iris, de las cuales 1 de las clases puede separarse linealmente de las otras, pero las otras 2 no son separables linealmente; por lo que es un ejemplo clásico y sencillo de clasificación multiclase. Además, al tener 4 features, el DM podrá reducir sus dimensiones y situar las muestras en un espacio 3D. Dicho esto, la librería scikit-learn tiene una función que permite obtener este Data Set, además de corregir 2 datos erróneos que tiene el repositorio de la UCI, por lo que se usará este conjunto para los experimentos [9].

Expuesto lo anterior, pasamos a realizar el pre-procesamiento de los datos. Primero eliminaremos los datos anómalos, para luego normalizar las muestras restantes con el estándar de media 0 y desviación 1. Una vez hecho esto, se separa las muestras restantes en 75% para Train y 25% para Test, respetando esta proporción para cada una de las clases. De esta forma, nuestros datos normalizados quedarán repartidos según la Tabla 4-3, estando listos para procesar.

	Nº de muestras (tras quitar anómalos)	Train	Test
Clase 0 (setosa)	49	37	12
Clase 1 (versicolor)	49	36	13
Clase 2 (virginica)	47	35	12
Totales	145	108	37

Tabla 4-3: Partición Train-Test del Data Set Iris

Ahora, empezaremos entrenado el DM con los datos de Train, utilizando los parámetros de la Tabla 4-4. Luego, usaremos Nyström para predecir la ubicación de los datos de Test. Por último, crearemos 3 SVMs, las cuales harán una regresión de los datos de Train para las coordenadas Ψ_1, Ψ_2 y Ψ_3 del DM de Train respectivamente. En la Tabla 4-5 encontramos

los hiperparámetros obtenidos por el cross-validation para el mejor modelo de cada una de las SVM.

Parámetro	Valor
Nº de autovectores computados (n_evecs)	10
Nº de vecino (k)	65

Tabla 4-4: Parámetros del DM Train de Iris

Coordenada DM (Ψ_i)	C	Épsilon	Gamma
Ψ_1	10000	0.1	'auto'
Ψ_2	100	0.001	'auto'
Ψ_3	10	0.001	1

Tabla 4-5: Hiperparámetros de las SVMs de Iris

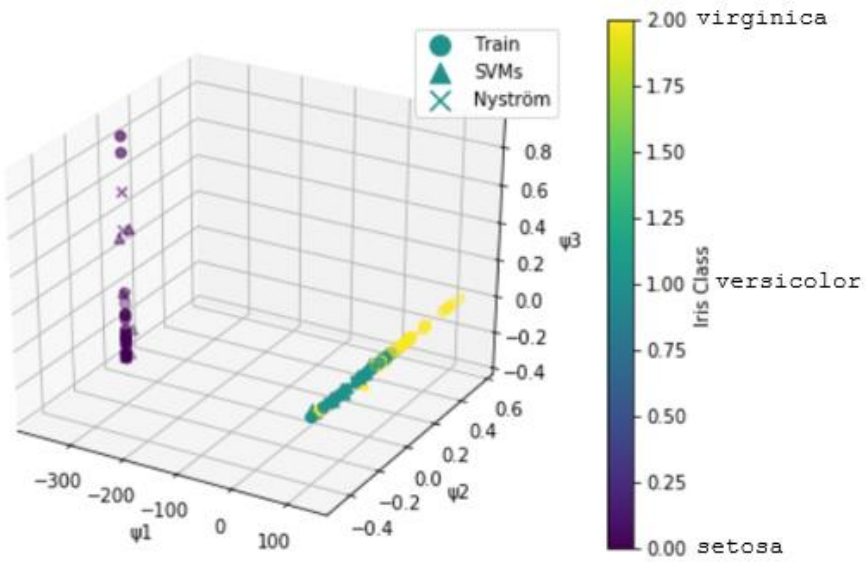

Figura 4-6: Gráfica Train-SVMs-Nyström de Iris (3 clases)

En la Figura 4-6 encontramos una gráfica con los datos de Train ubicados por el DM con los datos de Test, ubicados tanto por el método de Nyström como por las SVMs. Aquí se puede apreciar que tanto Nyström como las SVMs logran ubicar los datos correspondientes a cada clase del Test junto a los datos del Train de la misma clase. También, en este gráfico se puede apreciar las características de las clases del Data Set Iris, donde tenemos 1 clase (setosa) que es linealmente separable de las otras 2 clases a simple vista; mientras que las otras 2 clases (versicolor y virginica) no son separables linealmente, lo cual se puede apreciar al haber muestras de la otra clase en medio de la agrupación de muestra de cada clase. Producto de esto, vamos a evaluar, por un lado, como ubica las SVMs los datos para la clase Setosa y por el otro lado, como lo hacen con las clases Versicolor y Virginica.

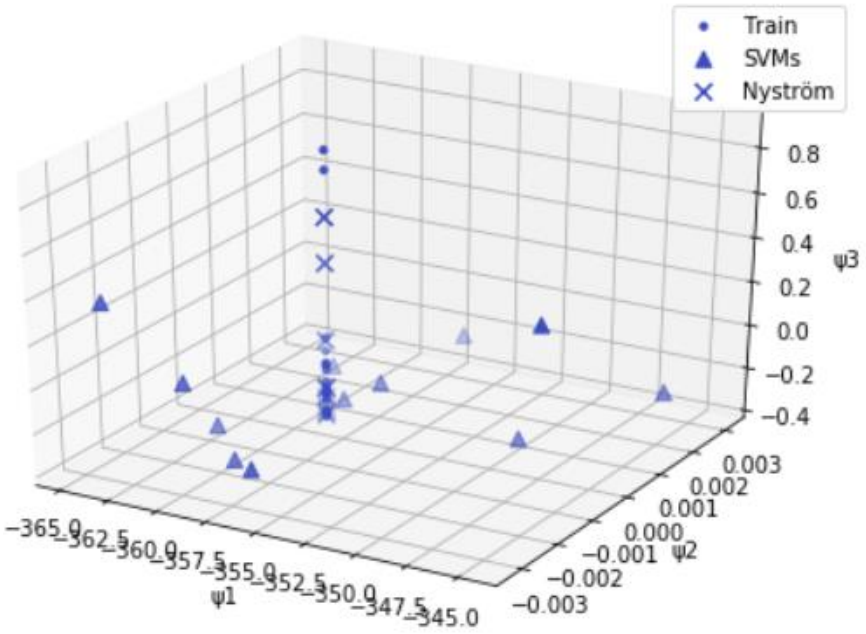

Figura 4-7: Gráfica Train-SVMs-Nyström de Iris (Setosa)

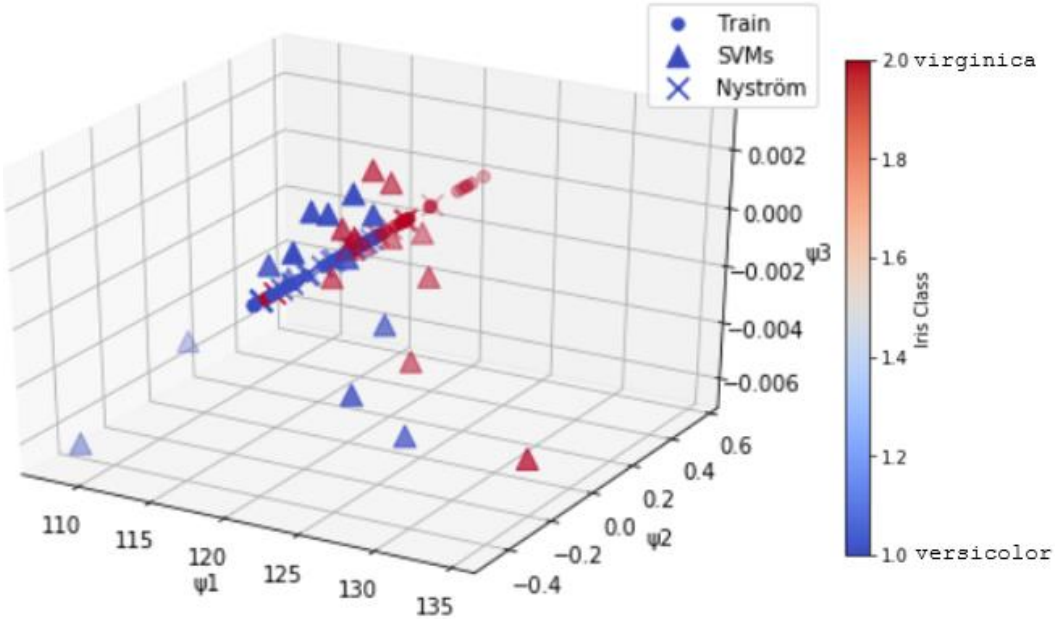

Figura 4-8: Gráfica Train-SVMs-Nyström de Iris (Versicolor-Virginica)

Empezando por clase setosa, representada en la Figura 4-7, podemos apreciar a simple vista que las predicciones de Nyström se encuentran casi superpuestas a los datos de Train; mientras que las predicciones de las SVMs se encuentran más dispersas. Si analizamos detenidamente, podremos notar que ambas predicciones tienen una pendiente similar en las coordenadas Ψ_2 y Ψ_3, siendo en la coordenada Ψ_1 donde difieren notablemente. Esto puede deberse tanto a la misma naturaleza de las SVMs de evitar el overfitting, pero al tener un relativamente bajo número de muestras, su regresión no es tan acertada. También es posible que, al ser esta una clase notablemente separable de las otras dos por medio de un corte lineal en esta coordenada, la SVM para Ψ_1 no sea demasiado precisa ubicando

internamente los datos para esta clase, por tener que contemplar también el posicionamiento de los datos de las otras dos clases. A pesar de esto, el resultado sigue siendo bastante aceptable para los datos de la clase setosa.

Por otro lado, tenemos las clases versicolor y virginica, representadas en la Figura 4-8. Aquí podemos apreciar un fenómeno parecido al de la clase Setosa, donde las predicciones de Nyström están casi sobrepuestas a los datos de Train; mientras que las predicciones de las SVMs parecen seguir la misma pendiente que los datos del DM, pero estas se encuentran más dispersas para la coordenada Ψ_1 y, algo más que para las setosa, para la Ψ_2. Todo esto se puede deber a lo explicado para la clase setosa, por lo que podríamos deducir que, si una de las clases se encuentra muy separada de las otras, es posible que la SVM, para la coordenada donde se encuentre el corte, tenga algunos problemas de precisión. Así mismo, esto último se podría tratar de solucionar aumentando la lista de opciones de cada hiperparámetro, aunque esto es muy costoso computacionalmente. No obstante, el resultado para ambas clases es bastante bueno, por lo que concluimos que las SVMs podrían ser un buen modelo para estimar las coordenadas de difusión de los datos out-of-simple en este ejemplo.

4.1 Experimento Wine

El siguiente conjunto de datos con el que trabajaremos serán el Data Set Wine. Este Data Set trata de 3 clases diferentes de vino italiano, pero a diferencia del Data Set del Iris, no se informa que alguna de las clases sea separable linealmente de las otras 2. Además, este Data Set cuenta con 13 features, por lo que es un ejemplo más interesante para ver como reduce dimensiones el DM que el Iris. Igual que para el Data Set anterior, la librería scikit-learn tiene una función que permite obtenerlo [9]. Cabe destacar que esta versión es una copia del conjunto de datos de UCI, variando en que esta última tiene el target como primera columna de cada muestra y que las clases están nombradas como 1, 2 y 3 [10]; mientras que scikit-learn las tiene como 0, 1 y 2 y ya tiene separado las muestras del target. Para este experimento se usó la versión de scikit-learn.

	Nº de muestras (tras quitar anómalos)	Train	Test
Clase 0	59	44	15
Clase 1	57	43	14
Clase 2	46	34	12
Totales	162	121	41

Tabla 4-6: Partición Train-Test del Data Set Wine

De esta forma, se pasó a realizar el pre-procesamiento de los datos. Primero eliminando los datos anómalos y normalizando las muestras restantes con el estándar de media 0 y desviación 1. Una vez hecho esto, se separaron las muestras restantes en 75% para Train y 25% para Test, respetando esta proporción para cada una de las clases. De esta forma, nuestros datos normalizados quedarán repartidos según la Tabla 4-6.

Parámetro	Valor
Nº de autovectores computados (n_evecs)	3
Nº de vecino (k)	64

Tabla 4-7: Parámetros del DM Train de Wine

Coordenada DM (Ψ_i)	C	Épsilon	Gamma
Ψ_1	100	0.01	0.01
Ψ_2	100	0.01	0.01
Ψ_3	1000	1e-06	0.01

Tabla 4-8: Hiperparámetros de las SVMs de Wine

Así pues, se pasó a entrenar el DM con los datos de Train, usando los de la Tabla 4-7. Luego, para predecir los datos de Test, se aplica Nyström sobre el DM con los datos de Train y se crean 3 SVMs, aplicando regresión sobre las coordenadas Ψ_1, Ψ_2 y Ψ_3 del DM de Train respectivamente. Los hiperparámetros obtenidos por cross-validation para las SMVs se encuentran en la Tabla 4-8.

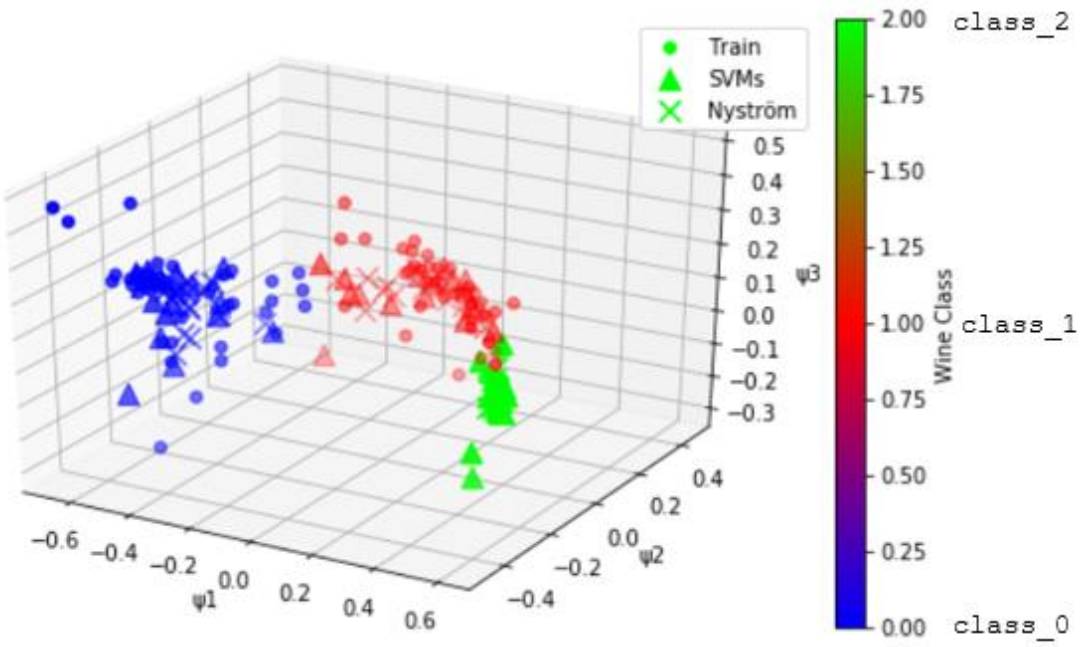

Figura 4-9: Gráfica completa Train-SVMs-Nyström de Wine

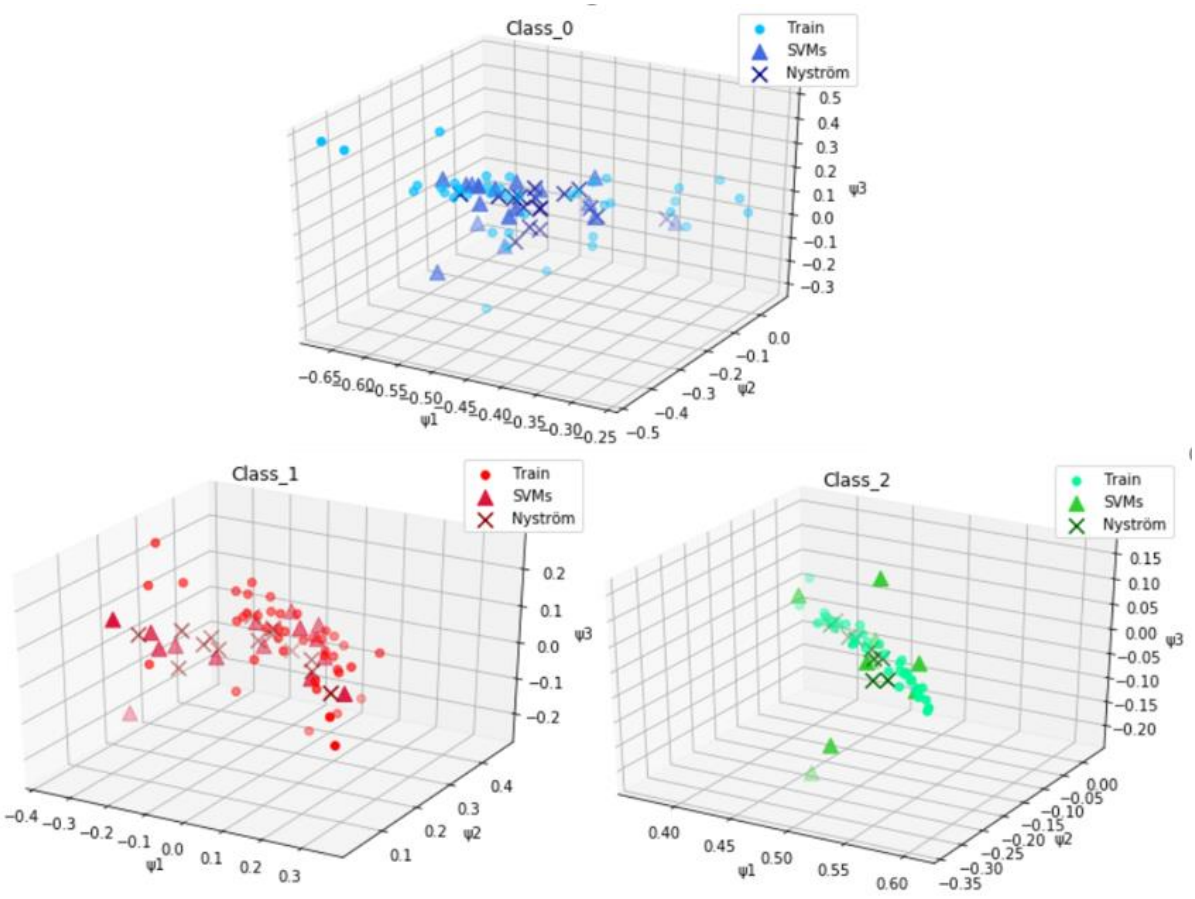

Figura 4-10: Gráfica Train-SVMs-Nyström de Wine (3 clases separadas)

En la Figura 4-9 se puede apreciar el posicionamiento de los datos de Train por parte del DM, así como las predicciones, tanto por Nyström como por SVMs, de los datos de Test. Claramente se puede apreciar que ubica correctamente los datos out-of-sample de la misma clase. Dado que las clases han formado agrupación, es interesante ver con detenimiento la ubicación de los datos obtenidos con las SVMs y compararlos con los obtenidos con Nyström.

De esta forma, tenemos la Figura 4-10, donde podemos apreciar las muestras, tanto del train como del test, por cada una de las clases. Empezando por la gráfica de la primera clase (Class_0), notamos que ambas predicciones dan un resultado bastante parecido, variando en que la predicción de la SVMs tiende a dispersarse un poco más. Esto mismo se puede apreciar en la gráfica de la segunda clase (Class_1), siendo ambas predicciones muy similares. Por último, en la gráfica de la tercera clase (Class_2), aquí podemos ver, en las predicciones de las SVMs, un resultado parecido al que daban en el Data Set de Iris, donde las muestras ubicadas por las SVMS tienden a dispersarse más que las ubicadas por Nyström. A pesar de esto, las predicciones de los datos de Test se encuentran correctamente ubicados con respecto a sus clases y a distancias razonables de los datos de Train, por lo que podemos afirmar que las SVMs hicieron un buen trabajo con los datos out-of-sample para este Data Set.

4.2 Experimento Breast Cancer

El último conjunto de datos con el que trabajaremos serán el Data Set Breast Cancer Wisconsin. Este Data Set almacena el diagnóstico de cáncer de mama de Wisconsin, clasificando las muestras en función de si el tumor es maligno o benigno. Así pues, tenemos un Data Set binario, que además tiene 30 features, por lo que es un conjunto de datos perfecto para aplicar DM. Como el Data Set anterior, la librería scikit-learn tiene una función que permite obtenerlo [9], siendo también una copia del conjunto de datos de UCI [10]. Para este experimento se usó la versión de scikit-learn.

De nuevo, pre procesamos el conjunto de datos quitando los datos anómalos y normalizando las muestras restantes con el estándar de media 0 y desviación 1. Para luego separar las muestras restantes en 75% para Train y 25% para Test, respetando esta proporción para cada una de las clases. De esta forma, nuestros datos normalizados quedarán repartidos según la Tabla 4-9.

	Nº de muestras (tras quitar anómalos)	Train	Test
Clase 0 (malignant)	159	119	40
Clase 1 (benign)	334	250	84
Totales	493	369	124

Tabla 4-9: Partición Train-Test del Data Set Breast Cancer

Hecho esto, pasamos a crear el DM con los datos de train, usando los parámetros de la Tabla 4-10. Para este experimento hemos usado un valor de k más pequeño que para los otros experimentos. Esto se debe a que la mejor distribución de los datos nos lo daba con k bajo, por lo que se puede concluir que este conjunto de datos da mayor importancia a los datos cercanos. También, la representación de los datos nos la da en espacio 2D, por lo que solo crearemos 2 SVMs para predecir las coordenadas Ψ_1 y Ψ_2. En la Tabla 4-11 se encuentran los hiperparámetros correspondientes al mejor modelo obtenido por cross-validation para cada una de las SVMs.

Parámetro	Valor
Nº de autovectores computados (n_evecs)	10
Nº de vecino (k)	5

Tabla 4-10: Parámetros del DM Train de Breast Cancer

Coordenada DM (Ψ_i)	C	Épsilon	Gamma
Ψ_1	10	0.1	0.01
Ψ_2	1000	1e-07	0.001

Tabla 4-11: Hiperparámetros de las SVMs de Breast Cancer

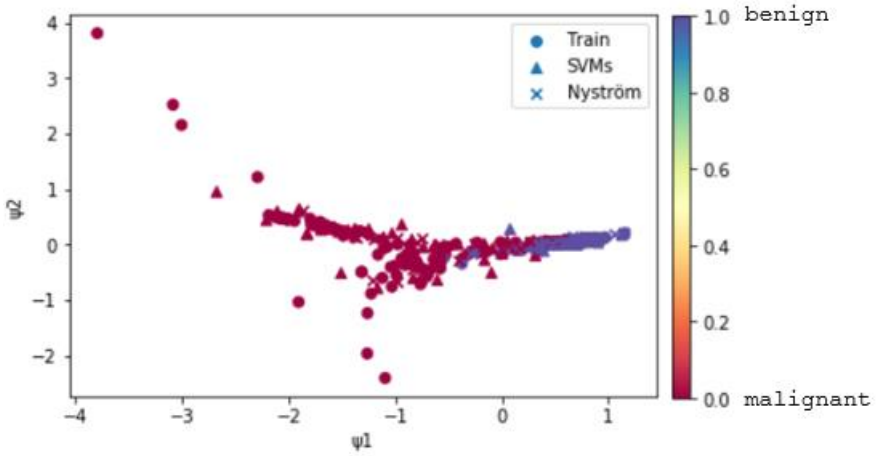

Figura 4-11: Gráfica completa Train-SVMs-Nyström de Breast Cancer

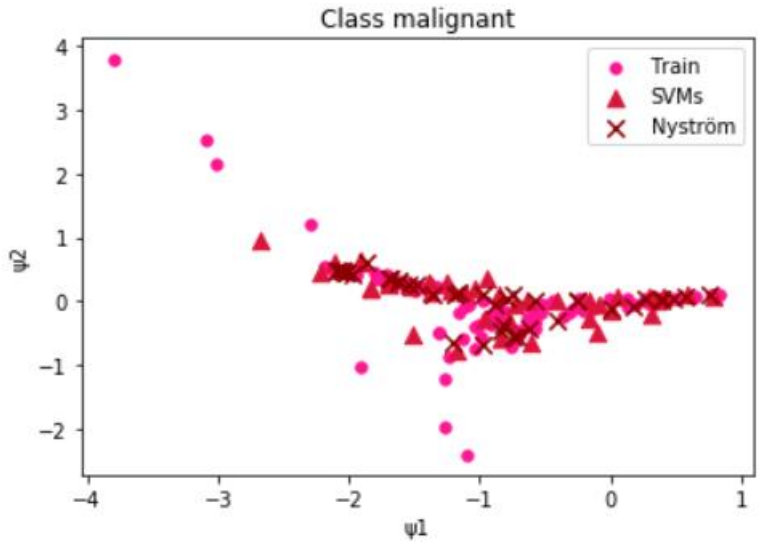

Figura 4-12: Gráfica Train-SVMs-Nyström de Breast Cancer (Malignant)

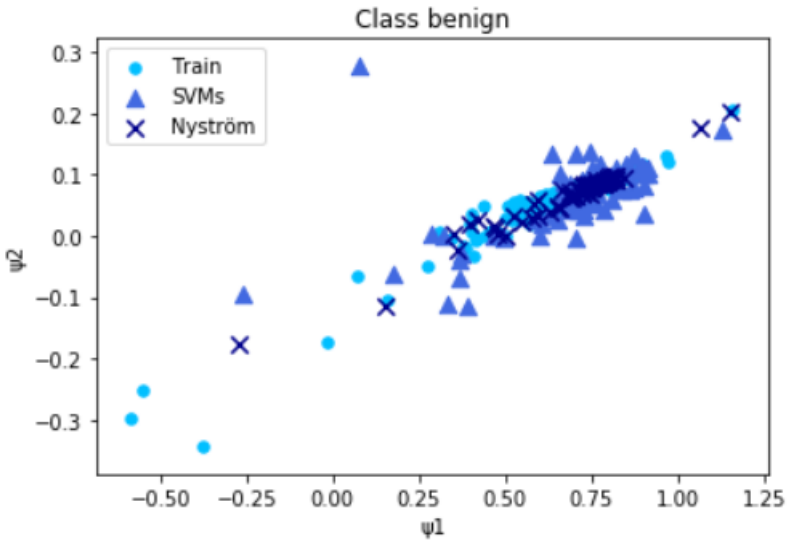

Figura 4-13: Gráfica Train-SVMs-Nyström de Breast Cancer (Benign)

En la Figura 4-11 podemos ver que los datos de train ubicados por el DM junto a los datos out-of-sample, en las ubicaciones obtenidas por Nyström y por las SVMs. De nuevo podemos apreciar que las SVMs ubican los datos en las zonas correspondientes a sus respectivas clases. Si analizamos los datos por clases, para clase malignant, Figura 4-12, podemos apreciar que ambas predicciones son muy parecidas, volviendo a presentarse que las SVMs tienen a predecir los datos algo más dispersos que Nyström. No obstante, en este caso es más notario que Nyström está sufriendo overfitting con respecto a los datos de train, ya que podemos observar que trata mucho de acercase a las zonas donde presentan más datos de train. Esto mismo se puede observar para la clase benign, graficado en la Figura 4-13. Así pues, podemos concluir que las SVMs hacen una buena estimación de las coordenadas para las muestras out-of-sample para este Data Set.

5 Conclusiones y trabajo futuro

5.1 Conclusiones

En los 4 experimentos realizados se puede notar que las SVMs no tienen problema alguno en ubicar los datos nuevos en las áreas correspondientes a las de sus respectivas clases. Incluso cuando se compara con otro método para ubicar datos out-of-sample, como lo es Nyström, podemos observar que los resultados obtenidos son muy similares. También podemos notar que las predicciones de las SVMs tienden, a diferencia de Nyström, a evitar el overfitting, como se muestra en los experimentos del Swiss Roll y el Breast Cancer. Por otro lado, las predicciones de las SVMs tienden a encontrarse más dispersas que las de Nyström para los experimentos del Iris y de Wine. No obstante, en estos experimentos se disponen de una muestra relativamente pequeña de cada clase con la cual entrenar, por lo que SVM no dispone de muestras suficientes para realizar la mejor regresión posible, así como, los DM no pueden realizar la mejor generalización de la representación de los datos. Esto último se puede comprobar con los otros experimentos, pues tienen una muestra de entrenamiento mayor y las predicciones parecen más acertadas.

De esta forma, podemos concluir que el uso de SVMs de regresión para obtener la función de cada coordenada del Diffusion Map al entrenarlos con los datos originales y el vector de la respectiva coordenada, parece una buena alternativa para solucionar el problema out-of-sample de los Diffusion Maps.

5.2 Trabajo futuro

Entre los trabajos a futuro proponemos la búsqueda de una métrica para poder cuantificar y comparar la calidad de las predicciones de las SVMs con respecto a Nyström y a las ubicaciones esperadas de esos datos dentro del Diffusion Map. Dado que DM es un método de aprendizaje no supervisado, no se puede obtener una ubicación exacta con la cual comparar directamente el resultado de las SVMs. Además, la forma del Diffusion Map entrenado con los datos de train puede llegar a variar completamente a la forma que ha recibido y ubicado todos los datos. Esto último abre la posibilidad de que las predicciones más dispersas que nos da las SVMs lleguen a ser más acertadas que las predicciones de Nyström, que tienden a seguir fielmente la forma de los datos in-sample del Diffusion Map.

Otra posible línea de trabajo futuro sería el comprender mejor el algoritmo de DM para poder optimizar sus parámetros, de igual forma a como se hace con los hiperparámetros de las SVMs. Tal que de esta forma se pueda sacar partido del hecho de que tanto SVM como DM utilizan un kernel para la representación de información.

Referencias

[1] Coifman, R.R. & Lafon, “Diffusion Maps”, Applied and Computational Harmonic Analysis, Elsevier, New York, 2004

[2] Fernández Pascual, Ángela, “Diffusion, methods and applications” (Tesis doctoral), UAM, Departamento de Ingeniería Informática, junio 2014

[3] Scholkopf, Bernhard & Smola, Alexander J., “Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond”, MIT Press, Cambridge, 2001

[4] Duda, Richard O., Peter E. Hart, & David G., Stork. “Pattern Classification” (Segunda Edición), Wiley-Interscience Publication, John Wiley and Sons, New York, 2001, pp. 259-265

[5] The SciPy community, “NumPy v1.16 Manual”, actualizado el 31 de enero de 2019, revisado el 10 de Junio de 2019, https://docs.scipy.org/doc/numpy/index.html

[6] Banisch, Ralf; Henning Thiede, Erik & Trstanova, Zofia, “pydiffmap”, Release 0.2.0.1, febrero 2019.

[7] Pedregosa *et al.*, “Scikit-learn: Machine Learning in Python”, Journal of Machine Learning Research, vol. 12, 2011, pp. 2825–2830.

[8] Marsland, Stephen, “Machine learning: an algorithmic perspective” (Segunda Edición), Chapman & Hall/CRC Machine Learning & Pattern Recognition Series, 2015, pp. 146-147.

[9] Scikit-learn developers, “Scikit-learn user guide”, Release 0.21.2, Mayo 2019, sitio web https://scikit-learn.org/stable/index.html , revisado el 10 de junio de 2019

[10] Dua, D. & Graff, C., “UCI Machine Learning Repository”, University of California, Irvine, School of Information and Computer Sciences, 2019, revisado el 10 de junio de 2019 http://archive.ics.uci.edu/ml

Glosario

Array	Lista o vector para almacenar datos.
Cross-validation	Técnica usada para estimar la precisión de un modelo, garantizando que los resultados sean independientes de la partición Train-Test
Data Set	Conjunto de muestras de datos
DM	Diffusion Map
Feature	Propiedad cuantificable o característica especifica de las muestras de un Data Set
Hiperplano	Objeto de N-1 dimensiones que divide un espacio de N dimensiones
In-sample	Dentro de la muestra original
Kernel	Núcleo
Out-of-sample	Fuera de la muestra original
Overfitting	Sobre ajuste, haber entrenado el modelo para que se asemeje mucho a los datos de entrenamiento.
SVM	Support Vector Machine
Target	Vector con los objetivos de las muestras de datos.